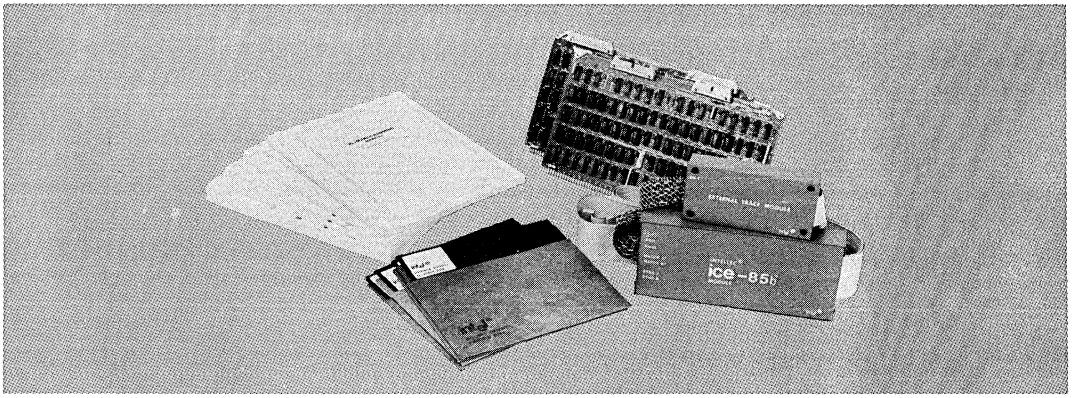
ICE-85B™
MCS-85™ IN-CIRCUIT EMULATOR
WITH MULTI-ICE™ SOFTWARE

- **Connects the Intellec® system resources to the user-configured system via a 40-pin adaptor plug**
- **Executes user system software in real-time (5 MHz clock)**
- **Allows user-configured system to share Intellec® memory and I/O facilities**
- **Provides 1023 states of 8085 trace data**
- **Displays trace data from the user's 8085 in assembler mnemonics and allows personality groupings of data sampled by the external 18-channel trace module**
- **Offers full symbolic debugging capability for both assembly language and Intel's high-level compiler languages PL/M-80 and FORTRAN-80**
- **The Multi-ICE™ software provides:**
  - **—for two In-Circuit Emulators to operate simultaneously in a single Intellec Microcomputer Development System.**
  - **—support for ICE 85/85™, 85/49™, and 85/41A™ Emulator combinations**
  - **—enhanced software features: symbolic display of addresses, macro commands, compound commands, software synchronization of processes, and INCLUDE file capability.**

The ICE-85B™ module resides in the Intellec® Microcomputer Development System and interfaces to the user system's 8085. It provides the ability to examine and alter MCS-85™ registers, memory, flag values, interrupt bits and I/O ports. Using the ICE-85B module, the designer can execute prototype software in real-time or single-step mode and can substitute Intellec® system memory and I/O for user system equivalent. ICE capability can be extended to the rest of the user system peripheral circuitry by allowing the user to create and execute a library of user-defined peripheral chip analyzer routines.

Multi-ICE In-Circuit Emulator is a software product which allows two Intel In-Circuit Emulators to run simultaneously in a single Intellec Microcomputer Development System. Multi-ICE software used in lieu of the standard ICE software gives users full control of the two ICE modules for debugging of multi-processor systems.



SYMBOLIC DEBUGGING CAPABILITY

ICE-85B allows the user to make symbolic references to I/O ports, memory addresses and data in his program. Symbols and PL/M-80 statement number may be substituted for numeric values in any of the ICE-85 commands. The user is relieved from looking up addresses of variables or program subroutines.

The user symbol table generated along with the object file during a PL/M-80 or FORTRAN-80 compilation or by the ISIS-II 8080/8085 Macro Assembler is loaded into the Intellec® System memory along with the user program which is to be emulated. The user may add to this symbol table any additional symbolic values for memory addresses, constants, or variables that are found useful during system debugging. By referring to symbolic memory addresses, the user can examine, change or break at the intended location.

ICE-85B provides symbolic definition of all 8085 registers, interrupt bits and flags. The following symbolic references are also provided for user convenience: TIMER, the low-order 16 bits of a register containing the number of 2 MHz clock pulses elapsed during emulation; HTIMER, the high-order 16 bits of the timer counter; PPC, the address of the last instruction emulated; BUFFERSIZE, the number of frames of valid trace data (between 0 and 1022).

PERSONALITY GROUPED DISPLAYS

Trace data in the 1023 by 42-channel real-time trace memory buffer is displayed in easy to read format. The user has the option to specify trace data displays in actual 8085 assembler instruction mnemonics. The data collected from the External Trace Module can be grouped and symbolically named according to user specifications and displayed in the appropriate number base designation. Simple ICE-85B commands allow the user to select any portion of the 42-bit trace buffer for immediate display.

MEMORY AND I/O MAPPING

Memory and I/O for the user system can be resident in the user system or "borrowed" from the Intellec® System through ICE-85B's mapping capability.

ICE-85B separates user memory into 32 2K blocks. Each block of memory can be defined independently. The user may assign Intellec® System equivalents to take the place of devices not yet designed for the user system during prototyping. In addition, Intellec® System memory or I/O can be accessed in place of suspect user system devices during prototyping or production checkout.

User ready synchronization—resource borrowing from the Intellec System is (at user option) independent of the user system; the user does not need to provide ready acknowledge when accessing resources mapped to the Intellec.

The user can also designate a block of memory or I/O as nonexistent. ICE-85B issues error messages when memory or I/O designated as nonexistent is accessed by the user program.

INTEGRATED HARDWARE/SOFTWARE DEVELOPMENT

The user prototype need consist of no more than an 8085 CPU socket and a user bus to begin integration of software and hardware development efforts. Through ICE-85B mapping capabilities, Intellec® System equivalents can be accessed for missing prototype hardware. Hardware designs can be tested using the system software which will drive the final product.

The system integration phase, which can be so costly when attempting to mesh completed hardware and software products, becomes a convenient two-way debug tool when begun early in the design cycle.

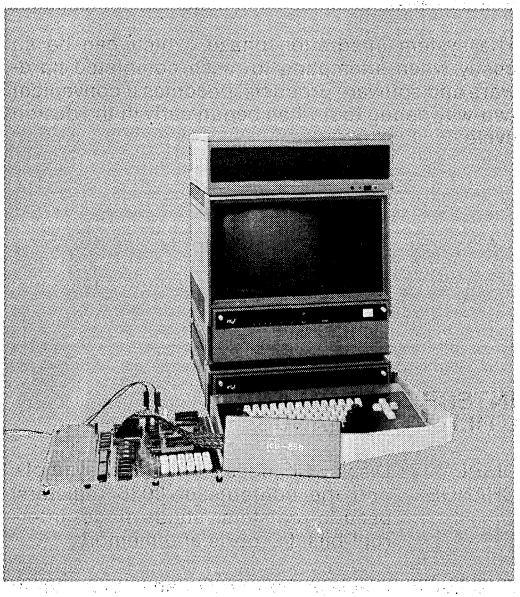
INTERROGATION AND UTILITY COMMANDS

DISPLAY/ CHANGE	Display/Changes the values of symbols and the contents of 8085 registers, pseudo-registers, status flags, interrupt bits, I/O ports and memory.
EVALUATE	Displays the value of an expression in the binary, octal, decimal or hexadecimal.
SEARCH	Searches user memory between locations in a user program for specified contents.
CALL	Emulates a procedure starting at a specified memory address in user memory.
ICALL	Executes a user-supplied procedure starting at a specified memory address in the Intellec® System memory.
EXECUTE	Saves emulated program registers and emulates a user-supplied subroutine to access peripheral chips in the user's system.

REAL TIME TRACE

ICE-85B captures valuable trace information from the emulating CPU and the External Trace Module while the user is executing programs in real time. The 8085 status, the user memory or port addressed, the data read or written, the serial data lines and data from 18 external signals, is stored for the last 1023 machine states executed (511 machine cycles). This provides ample data for determining how the user system was reacting prior to emulation break. It is available whether the break was user-initiated or the result of an error condition.

For detailed information on the actions of CPU registers, flags, or other system operations, the user may operate in single or multi-step sequences tailored to system debug needs.



EMULATION CONTROLS AND COMMANDS

GROUP — Defines into a symbolically named group, a channel or combination of channels from the 8085 Microprocessor and/or the External Trace Module.

GO — Initiates real-time emulation and controls emulation break conditions.

STEP — Initiates emulation in single instruction steps. User may specify the type and amount of information displayed following each step, and define conditions under which stepping should continue.

PRINT — Prints the user-specified portion of the trace memory to the selected list device.

EXTERNAL TRACE MODULE

TTL level signals from 18 points in the user system may be synchronously sampled by the External Trace Module and collected in ICE-85B's trace buffer. The signals can be collected from a single peripheral chip via the supplied 40-pin DIP clip or may be placed by the user on up to 18 separate signal nodes using the supplied 18 individual probe clips. These signals are included in the 42-channel breakpoint comparisons and clock qualifiers. Also, data from these 18 channels may be displayed in meaningful, user-defined groupings.

SYNCHRONOUS OPERATION WITH OTHER DESIGN AIDS

ICE-85B can be synchronized with other Intellec® design aids by means of two external synchronization lines. These lines are used to enable and disable ICE-85B trace data collection and to cause break conditions based on an external signal which may not be included in the ICE-85B breakpoint registers. In addition, ICE-85B can generate signals on these lines which may be used to control other design aids.

BREAK REGISTERS/ TRACE MEMORY

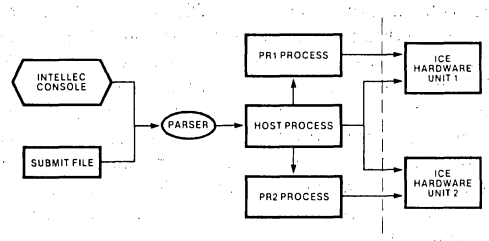
ICE-85B has two breakpoint registers which are used to break emulation, and two trace qualifier registers which are used to control the collection of trace data during emulation. Each register is 42 entries wide, one entry for each channel and each entry can take any one of the three values 0, 1 or "don't care."

The trace buffer, also 42 entries wide, collects data sampled from 24 8085 processor channels and 18 external channels sampled by the External Trace Module. The signals collected from the 8085 include address lines, data lines, status lines and serial input and output lines. The 18 channels extending from the External Trace Module synchronously sample and collect into the trace buffer any user-specified TTL compatible signal from the rest of the prototype system. "Break" and "trace qualification" may therefore occur as a result of a match of any combination of up to 42 channels of CPU and external circuitry signals.

MULTI-ICE™ OPERATION

Multi-ICE software is a debug tool which allows two ICE emulators to begin and stop in sequence. Once started, two ICE emulators emulate simultaneously and independently. Thus, Multi-ICE software permits the debugging of asynchronous or synchronous multi-processor systems.

A conceptual model for the Multi-ICE software can be illustrated with the following block diagram.



Block Diagram of Multi-ICE™ Operation

There are three processes in the Multi-ICE environment: the Host process and the two ICE processes to control the two ICE hardware modules. The processor for these three processes is the microcomputer in the Intellec Microcomputer Development System. Only the Host process is active when Multi-ICE software is invoked. The Parser interfaces with the console, receives commands from the console or from a file, translates them into intermediate code, and loads the code into the Host command code buffer or ICE command code buffers.

The Host process executes commands from its command code buffer using the execution software and hardware of the Host's current environment, either environment 1 or environment 2 (EN1 or EN2), as required. EN1 and EN2 are the operating environments of the two In-Circuit Emulators.

The user can change the execution environment (from EN1 to EN2 or vice versa) with the SWITCH command. Once the environment is selected, ICE operation is the same as with standard ICE software. In addition, the enhanced software capabilities are available to the user.

The two ICE processes (PR1 and PR2) execute commands from their command code buffers in their own environments (PR1 in EN1 and PR2 in EN2). The main functions of the two ICE execution processes are to control the operations of the two ICE hardware sets. The ACTIVATE command controls the execution of the ICE processes. Commands are passed on to each ICE unit to initiate the desired ICE functions.

The two ICE hardware units accept commands from the Host process or ICE processes. Once emulations start, the two ICE hardware sets will operate until a break condition is met or processing is interrupted by commands from the ICE execution processes.

Symbolic Display of Addresses

The user has the option of displaying a 16-bit address in the form of a symbol name or line number plus a hex number offset.

Macro Command

A macro is a set of commands which is given a name. Thus, a group of commands which is executed frequently may be defined as a macro. Each time the user wants to execute that group of commands, he may just invoke the macro by typing a colon followed by the macro name. Up to ten parameters may be passed to the macro.

Macro commands may be defined at the beginning of a debug session and then can be used throughout the whole session. If the user wants to save the macros for later use, he may use the PUT command to save the macro on diskette, or the user may edit the macro file off-line using the Intellec text editor. Later, the user may use the INCLUDE command to bring in the macro definition file that he created.

Example:

```
*DEFINE MACRO INITMEM     ;This macro clears the memory and then loads the programs.
.*SWITCH = EN1            ;Select environment 1 (ICE Module 1)
.*BYTE 0 TO 100=0         ;Initialize memory to 0.
.*LOAD:F1:DRIVER          ;Load user program into memory for ICE Module 1.
.*SWITCH = EN2            ;Select environment 2 (ICE Module 2)
.*LOAD:F1:DR2             ;Load user program into memory for ICE Module 2
.*EM                      ;End of Macro
*                         ;To execute this Macro, user types :INITMEM
```

Compound Command

Compound commands provide conditional execution of commands (IF Command) and execution of commands repeatedly until certain conditions are met (COUNT, REPEAT Commands).

Compound commands and Macro commands may be nested any number of times.

Example:

```
*DEFINE .I = 0             ;Define symbol .I to 0
*COUNT 100H                ;Repeat the following commands 100H times
.*IF .I AND 1 THEN         ;Check if .I is odd
..*BYT .I = .I             ;Fill the memory at location .I to value .I
..*END
.*.I = .I + 1              ;Increment .I by 1
.*END                      ;Command executes upon carriage-return after END
```

INCLUDE File Capability

The INCLUDE command causes input to be taken from the file specified until the end of the file is encountered, at which point, input continues to be

taken from the previous source. Nesting of INCLUDES is permitted. Since the command code file can be complex, the ability to edit offline becomes desirable. The INCLUDE command allows the user to pull in command code files and Macro commands created offline which can then be used for the particular debugging session.

Example:

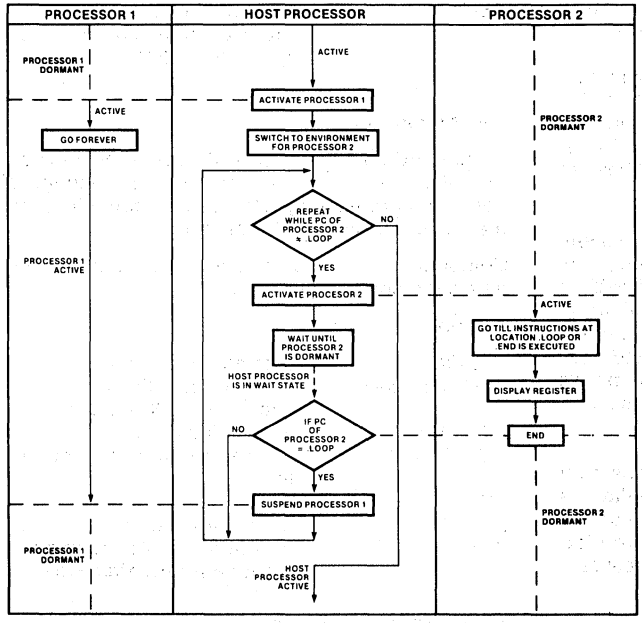
```
*INCLUDE :F1:PROG1    ;Cause input to be taken
                      from file PROG1
*MAP 0 LENGTH 64K     ;Contents of the file
=USER                 PROG1 are listed on
                      screen as they are
                      executed.

*MAP IO 0 TO FF
=USER
*SWITCH = EN2
*LOAD :F2:LED.HEX
*SWITCH = EN1         ;End of the file PROG1
*                     ;After the end of file is
                      reached, control is
                      returned to console.
```

Software Synchronization of Processes

Up to three processes (Host, PR1 and PR2) can be active simultaneously in the system. An ICE process can be activated (ACTIVATE), suspended (SUSPEND), killed (KILL), or continued (CONTINUE). The Host process can wait for other processes to become dormant before it becomes active again. Through these synchronization commands, the user can create a system test file off-line yet be able to synchronize the three processes when the actual system test is executed.

Example:

The capability of the software synchronization commands is demonstrated by the following example. The flowchart shows the synchronization requirements. The program steps show the actual implementation.




```
*ACTIVATE PR1          ;Activate PR1
.*GO FROM 800          ;Start ICE Module 1
.*END                  ;End of Activate block
PR1 EMULATION BEGUN
*SWI=EN2               ;Switch execution Environment to EN2
*REPEAT                ;Repeat the following block of
                        commands while PC is not equal to .Loop
.*WHILE PC < > .LOOP
.*ACT PR2              ;Activate PR2
..*GO TILL .LOOP OR .END ;Go till instruction at location .Loop
                        or at location .END is executed
..*REGISTER            ;Display the registers
..*END                 ;End of Activate block
.*WAIT PR2             ;Wait until PR2 is dormant
.*IF PC=.LOOP THEN
..*SUSPEND PR1
..*END                 ;End of IF block
.*END                  ;End of REPEAT block
*
```

Flowchart of the Example for Demonstrating Multi-ICE™ Synchronization Capability

SPECIFICATIONS

ICE-85B™ Operating Environment

Required Hardware:
- Intellec® Microcomputer Development System
  - (64K bytes RAM for Multi-ICE software)
  - (32K bytes RAM single ICE software)
  - System Console
  - Intellec® Diskette Operating System
  - ICE-85B Module

Required Software:
- System Monitor
- ISIS-II
- ICE-85B or Multi-ICE Software

Equipment Supplied

18-Channel External Trace Module
Printed Circuit Boards (2)
Interface Cable and Emulation Buffer Module
Operator's Manuals
ICE-85B Software
Multi-ICE Software
- Contains software that supports 85/85 Emulators, 85/49 Emulators and 85/41A Emulators

Emulation Clock

User's system clock or ICE-85B adaptor socket (10.0 MHz Crystal)

Physical Characteristics

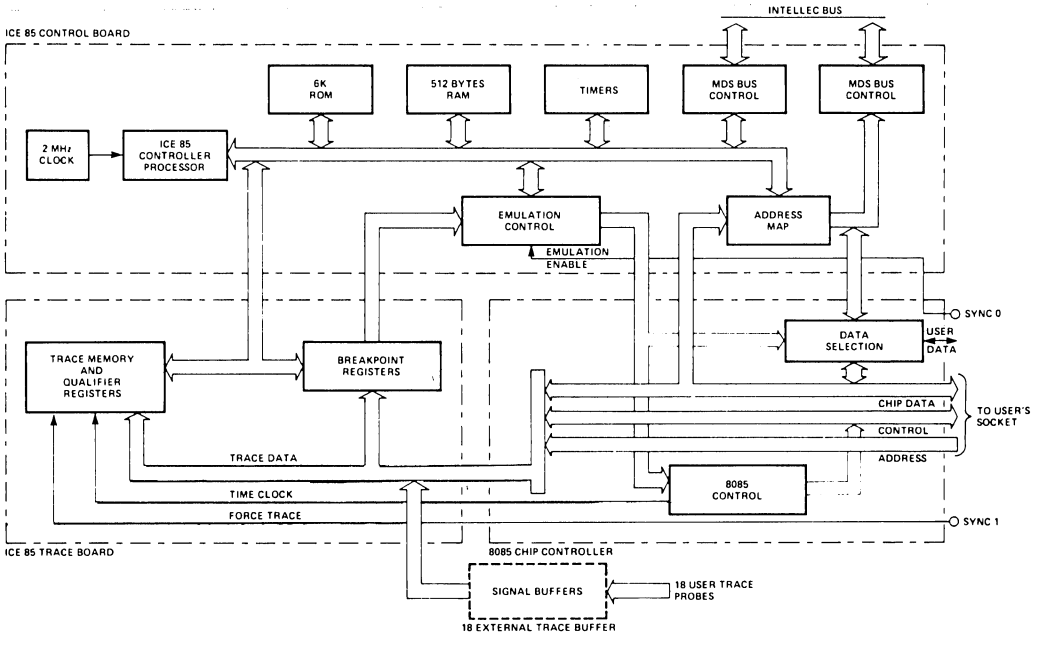
Printed Circuit Boards:
- Width: 12.00 in. (30.48 cm)
- Height: 6.75 in. (1715 cm)
- Depth: 0.50 in. (1.27 cm)
- Packaged Weight: 6.00 lb (2.73 kg)

Electrical Characteristics

DC Power:
- V_{CC} = + 5V ± 5%
- I_{CC} = 12A maximum; 10A typical
- V_{DD} = + 12V ± 5%
- I_{DD} = 80 mA maximum; 60 mA typical
- V_{BB} = − 10V ± 5%
- I_{BB} = 1 mA maximum; 10 μA typical

Environmental Characteristics

Operating Temperature: 0° to 40°C
Operating Humidity: Up to 95% relative humidity without condensation.



ICE-85B™ BLOCK DIAGRAM

Ordering Information

Part Number	Description
MDS*-85B-ICE	8085 CPU In-Circuit Emulator, 18-Channel External Trace Module and Multi-ICE software
MDS*-85U-ICE	Upgrade kit to convert ICE-85 or ICE-85A to ICE-85B functionality. Consists of Multi-ICE software and 5MHz Hardware

*"***MDS***"* is an ordering code only, and is not used as a product name or trademark. MDS® is a registered trademark of Mohawk Data Sciences Corp.